**Solution Design for Smarter Reconciliation and Anomaly Detection**

**1. High-Level Architecture:**

* **Data Ingestion Layer:** Load and preprocess reconciliation data from CSV/Excel.
* **Anomaly Detection Layer:** Identify and classify anomalies using ML techniques and historical data.
* **LLM Integration:** Use LLMs like GPT or LLaMA for generating break resolution summaries and providing insights.
* **Feedback & Learning Module:** Capture user feedback on anomalies (false positives/negatives) for model improvement.
* **Automation & Workflow Layer:** Automate break resolution actions — API calls, emails, ticketing.
* **Interactive UI:** Provide an interface for reconcilers to view and interact with detected anomalies.

**2. Detailed Component Breakdown:**

|  |  |  |
| --- | --- | --- |
| **Component** | **Purpose** | **Tools/Techniques** |
| **Data Ingestion** | Load and preprocess reconciliation data | Pandas, NumPy |
| **Anomaly Detection** | Identify outliers, classify anomalies | Scikit-learn (clustering, anomaly detection) |
| **LLM Integration** | Generate resolution summaries, insights | OpenAI GPT, Hugging Face Transformers |
| **Feedback & Learning** | Capture feedback to improve models | Streamlit/Gradio for feedback capture |
| **Automation & Workflow** | Create tasks, trigger APIs, notifications | Requests (API calls), Jira API, SMTP |
| **Interactive UI** | Display results, capture feedback | Streamlit, Gradio, Plotly for visualizations |

**3. Solution Flow:**

1. **Data Loading:** Read reconciliation data (CSV/Excel) and clean/preprocess it.
2. **Anomaly Detection:** Use clustering (K-means, DBSCAN) and anomaly detection techniques (Isolation Forest, Z-score).
3. **Classification:** Classify anomalies based on predefined buckets; unknown reasons are marked as "new."
4. **LLM for Summarization:** Generate concise break resolution summaries using LLMs.
5. **Feedback Mechanism:** Capture user feedback for model improvement.
6. **Automation:** Create tasks, trigger API calls, send emails, or create tickets.
7. **UI and Visualization:** Display anomaly results and capture feedback.

**4. Tech Stack & Libraries:**

* **Python Libraries:** Pandas, NumPy, Scikit-learn, Requests, Plotly, Streamlit, Gradio.
* **ML & AI:** OpenAI GPT, Hugging Face Transformers for LLMs.
* **Automation:** Jira API, SMTP for email, REST APIs for integrations.